14 | 故事案例（下）：小步快跑，小而美的敏捷

2019-11-28 雷蓓蓓

项目管理实战20讲 进入课程 >

讲述：雷蓓蓓
时长 14:14　大小 13.04M

你好，我是雷蓓蓓。今天，我们来聊一聊敏捷。

很多人认为，每天开站会，有固定时长的迭代，有自己的“Scrum Master”，就是在开展敏捷实践了，其实不然。

具备敏捷之形的团队有很多，但是，真正掌握敏捷精髓的，却并不多见。这是因为，敏捷方法属于 simple but not easy（简单但并不好做）。结合我这么多年的体会来看，与其说敏捷是一场研发方式的变革，不如说是一场思维方式的变革。

> 图片来源：https://agilemanifesto.org/

今天，我会结合我在某试点团队中深度实践敏捷的经历，来跟你分享一下，我对敏捷精神的理解，以及在敏捷应用过程中的实施建议。

为什么引入敏捷？

敏捷的特点，就是小即是美（Small is beautiful）。这个小而美，体现在人、事、时间三个方面：

1. 人：拆分成小规模（5～7 人）、跨职能的小团队；
2. 事：拆分成一系列小而具体的交付物，按优先级排序，增量交付；
3. 时间：拆分成固定大小的短迭代（1～4 周），在每个迭代结束后，对可工作的产出进行演示。

总体来说，就是用小团队在小块时间，做出小块的东西来，并且周期性地集成组装。为什么我们当时会考虑引入敏捷呢？这就要从第一个版本的发布讲起了。

这个新业务的第一个版本，原本预计在春节前发布。我们基于 WBS 做了完整的项目计划，用两个月时间进行模块开发，然后用一个月的时间来做发布前的联调、功能及性能测试。

开发联调开始之前，一切都非常理想。我们有自认为完备的计划，有周会、周报和各种文档，还有任务系统的监控报表。我们从种种途径获取到的信息，都在告诉我们：“一切正常！”

直到有一天，我们开始代码集成，准备测试了，“嘭！”各种意外超出了我们的想象，项目越来越不可控：

集成联调比我们想像得要更复杂；

性能稳定性调优花了更多的时间；

有一名主要开发人员离职；

这期间赶上了过年，要放十来天假；

测试人员是个新人，还在学习期

……

所有因素加在一起，把这个项目拖入了完全失控的深渊，一直到“三·八”妇女节才正式提交给了用户……

最初引入敏捷，原因很简单，就是想要**发布周期更短。**

痛点一：发布时间不可控（快速的增量交付）

考虑到项目的实际背景，我们把迭代周期定为一个月。我们每个月都会跟内部客户一起做 Sprint 计划及 Review 演示。这种做法，给我们带来了哪些改进呢？

提早集成与测试。这让问题得以及时暴露，带来了更快的反馈及应对；

及时规避风险。意外仍然会有，但大多数情况下，我们可以在 Sprint 内部消化掉，避免更大的影响；

快速响应变化。在 Sprint 1 演示会后，我们收到了新客户提的紧急需求，立即做了相应的调整。如果按照之前的模式，这个时候，可能很多事情我们都只做了一半，想调头没那么容易。短周期让我们可以灵活调整方向，每个 Sprint 都是**潜在可交付**的产品。

另外，在 Sprint 3 之后，我们临时安排了一个小的 Sprint，用来**快速地将“潜在可交付”变为“真正可交付**”。我们发现，在每个周期内，能够真正把事情做完，跟我们的最终用户一起分享阶段性成果，对团队来说，也是非常好的激励。

这时，发布周期的问题已经基本解决了，我们的交付灵活性高了很多，客户也更满意了。那么，我们是否可以号称是个 Scrum 团队了呢?

痛点二：摆脱“接力综合征”（从对抗走向协作）

经过仔细地观察和总结，我发现，团队各个角色之间的协作方式仍然存在着一些问题，我把这叫作“接力综合征”。虽然交付周期变成了每月一次，但大家却仍然在按照过去的方式工作。具体表现有以下两点：

1. 宁愿选择等待

每个人都等着上一环节的人把东西弄好送到自己面前来，才能开始工作。比如，我经常听到这样的说法：“需求文档还没理清楚，急啥？”“接口设计文档还没确认，怎么做啊？”这种情况，在传统的项目管理中是很正常的。但是，这些空耗的时间，并不能给产品带来直接的价值。

2. 角色间泾渭分明

每个人都觉得，只要把自己份内的事做完就行了。比如，开发把工作做完了，也不管做得效果咋样，心想：“反正要丢给测试，我先撤了，测出问题再说。”测试测出来 Bug，心想：“等开发全部修完，我再接着测，反正我都测完了。”

在这种情况下，各角色之间是有一定的对抗的。项目中的任何一件小事都有可能造成冲突。最终大家都耗在那儿，每个人更在意的是“这是你的问题，不是我的问题”，却没有把焦点放在达成整体目标上。

在传统的项目管理中，项目经理的很大一部分工作就是要厘清各方责任，界定权利与义务，疏通对抗情绪，并解决随之而来的突发问题。但在敏捷的情境中，更加快速的交付压力，使得这种等待和界限变得越来越不可接受，我们不得不改变思路。

敏捷的打法，就好比是打橄榄球，所有队员都时刻关注着场上的比分。虽然彼此之间有分工，但作为一个 team，进球才是最关键的。敏捷也是一样。我们从敏捷思想中得到启发，开始进行一系列的改进。

首先，开发和测试把位置搬到了一起，并且设定了**共同的 Sprint 目标和完成标准。**

开发做完了工作以后，如果发现测试进度被卡，就会跟着一起着急，一起想办法解决问题，因为这影响到了整体的进度。

其次，**从“你完成 - 我开始”，到我们一起完成。**

在敏捷团队中，开发干得热火朝天的时候，测试也没闲着，测试代码与开发代码几乎是同时在写的。往往代码刚“出炉”就测上了，而且只有在测试结束并确认没有 Bug 的时候，开发的工作才算结束。

我们使用故事点燃尽图，来衡量整体进度的偏差，并且团队会约定好，只有一个功能点完全测试通过，燃尽图才能往下走。这个燃尽图成了团队的计分牌，每个人都关注着同一个目标。

同时，我们还发明了里程碑燃尽图，用来衡量每个迭代对于整体进度的贡献，以及多个迭代之间累积需求总量的变化，相当于一个赛季的累积记分牌。我给你分享一份**燃尽图型项目进度模版**，你可以点击网盘获取，提取码是 pmx8。

这些措施打破了角色之间相互切分和推诿的局面，共同的目标让我们变成了一个价值共同体。**毕竟，只有协作，才能达成目标。**

痛点三：需求理解不一致 （面对面澄清及估算）

至此，团队的力量得到了很好的凝聚。在复盘会上，大家畅所欲言，共同讨论了下一个待改进项，那就是是需求理解。这应该是技术类项目的一个共性问题。

当时，我们团队没有专职的策划，开发人员在理解需求时，经常会感到非常困难。我们打开敏捷宝箱，找到了一条重要的价值观“**个体与交互 > 过程与工具**”。相比于更多、更长的需求文档，我们决定采用更多的面对面交流来解决这个问题！

于是，我们把迭代计划会分成了两个部分：

1. 产品负责人向团队和用户代表，面对面地讲解收集来的各方需求，最终明确需求的优先级及验证条件，也就是说，在迭代结束的 Demo 演示会上，我们要给用户呈现什么。
2. 团队估算。我们采用敏捷估算扑克的形式，由团队成员共同给出估算结果，最后综合得出这个迭代要交付哪些内容。

团队估算的过程，其实是个双向互动的环节，可以帮助团队和产品负责人共同加深对条目的理解。同时，产品负责人也会根据大家的反馈，及时修改条目，完善条目。

具体估算时，为了避免干扰估算结果，当所有成员选择好代表自己估算值的纸牌后，大家同时亮牌。同时亮牌的好处是，**不会有人跟风出牌，每个人的估算都是经过独立思考得出的**，这也是扑克估算的精华所在。

如果估算值差距明显，代表大家对该条目的工作量没有获得共识，团队需要对该条目的评估结果进行讨论，由最大值和最小值的牌主，分别说明自己的估算理由，并重新讨论，确定最终的评估结果。

经过实践检验，这样面对面的需求沟通及评估，至少带来了以下三方面的好处：

1. 需求探索更深入。

在计划会上，团队会直面一线用户，需求可以得到面对面的交流和澄清。另外，团队估算其实也是一个团队共同探索需求的过程。因为只有到真正考虑要花多少成本去做的时候，才会有各种各样的问题暴露出来。这个方法对于在早期进一步挖掘需求细节，特别有帮助。

2. 估算结果更加全面、细致。

在传统的项目管理中，我们也做估算。比如，WBS 分配好了任务，然后每个人估算自己的。因为每个人都只对自己的那部分任务比较熟悉，估算时往往会有欠考虑，而团队估算，就很好地补足了这一点。

每一个故事都会由全员出牌，各方从自己的角度出发想问题，可以互相补充。比如，在估算时，测试的成本也会被考虑进去。对于测试成本高的功能点，开发会主动想办法加强单元测

试等白盒测试手段，这样一来，估算结果自然更细致、全面。

3. 找到了更优的整体解决方案。

由于各方共同参与估算，前端、中间层、后端、测试的思路可以在一起交流碰撞，这样就非常利于找到最优的解决方案。

我们团队的这一系列敏捷尝试，始终都是围绕着项目中切实的痛点展开的。从最开始缩短发布周期、经常交付可工作的软件，到应对“接力综合征”，提升团队的整体目标感和协作效率，再到探索更加有效的需求理解及团队估算方式，在增进团队交流的同时，又保障了需求质量。

敏捷实践的应用，为我们带来了诸多好处：

1. 提升客户体验

更低的延误率

阶段性可见的产出

更快的反馈、适应与调整

2. 提升管理者体验

团队自主运行，管理更轻松

变“赶”为“引”，为共同目标奋斗

3. 提升团队体验

更高的生产力

更强的责任感、主人翁意识

总结

今天，借助一个实际案例，我跟你分享了我们在应用敏捷方法的过程中，对于敏捷思想的体会和运用。

你可以看到，对于敏捷方法，我们并不是拿来即用的。我们所采用的这些方法，大多是以敏捷思想为指导，以敏捷方法为基础，在实际场景中不断演化，一点点改进出来的。实际上，没有任何一种方法、工具可以放之四海而皆准，每个人都需要在自己的场景中思考。

真正决定一个团队是否敏捷的，不在于是否应用了那些实践，而在于实践背后是否体现了敏捷精神。通过我们的长期实践和观察理解，我提炼出了实战中三项最重要的敏捷精神，那就是：**快速可靠交付，用户价值驱动，持续自发改进。**

我希望你能坚持敏捷精神，而不是僵化地套用特定的做法。在团队中实践应用敏捷时，也应该遵循小而美的原则，每次一小步，挑一个痛点去集中解决，小步快跑，不断尝试和优化。只要你做到了以上三点敏捷精神，那么，你的团队就是一个敏捷的团队，你的组织就是一个敏捷的组织。

畅所欲言

最后，我想请你做个自检：你所在的团队，有哪些敏捷实践已经偏离了敏捷的初衷？从敏捷精神出发，你还可以找到哪些小而美的好方法？

欢迎你畅所欲言，我在留言区等你，也欢迎你把文章分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　13 | 故事案例（上）：新手上路，如何引入变化?

下一篇　15 | 工具方法串讲：手把手教你高效管理

精选留言 (10)

写留言

K战神
2019-11-28

最近我们部门甚至说公司层面在试点敏捷。

我也从中一直探索好的方式，领悟其精髓。上周六我还写了一篇总结。https://www.cnblogs.com/sunchong/p/11917766.html
...
展开

作者回复: 谢谢！赞一下从头听到尾的好同学

组织架构的影响要更大些，我了解到的互联网公司，多是弱矩阵式架构，产品和用户探索的不确定性显著增加，这跟传统型项目有很大不同，很少是单个PM就能完全负责实施交付的。整体来看，方法还是万变不离其宗，但是对项目经理的软实力要求更高了，挑战更大！

2 2

穷查理
2019-12-02

想借用孙子的一句话赞颂一下敏捷： “兵无常势,水无常形,因敌变化而取胜者,谓之神。”
方法都在那里，关键还是要看因时因地的运用~

展开

Jason
2019-12-01

我，一枚程序员，偷偷看到现在，没敢说话。但实在忍不住了，说一句:讲的真不错!

作者回复: 哈哈哈，千万别憋着.....

谢谢你的赞美，我会继续努力哒!

like_jun
2019-12-01

交付是团队的事情。而不是个人的事情。技能不同负责的工作会不同。但这不能影响团队交付，要齐心协力。才能把交付做好。

展开

作者回复: 感谢总结~

maks
2019-11-29

使用小步快跑的理念,结合上一讲中的"地利",参考下蓓格格的最佳实践.
可以得出适用于当前环境的敏捷方法. (0_0)
说来惭愧,刚听完这一讲中的橄榄球式为比分全局思考, 不要局限与个人任务的责任理念
结果工作的时候,就犯了这一点,事情是这样的:
上午 业务人员跟我说出了问题,然后我一看 是手机App端的...

展开

作者回复: 这是一种惯性的自我保护思维，角色间自然形成的边界，也很正常。项目管理做久了，就会生出一种无边界感，最好的锻炼，就是留意自己考虑问题的视角，要一直放在整体目标上。你可以在每一次特别事情的处理上，像今天这样就很棒，建立自检的习惯，多给自己复盘。

有心猴
2019-11-28

敏捷解决的是需求太多时沟通成本的问题,而不是那种说了不算,算了不说,天天改主意的问题

许童童
2019-11-28

快速可靠交付，用户价值驱动，持续自发改进。理解了这三点，也就领会了敏捷开发的精髓。

作者回复: 👍

陈宇明
2019-11-28

后续会有使用敏捷的小团队的项目案例吗?

展开

作者回复: 这个就是啊！具体是指哪方面?

1

Cy23
2019-11-28

划分小团队可以揪出混子，才分项目定时定量快速推进，挺好，总比一下子一堆干起来积极，分批快速交付，循序渐进，时间拆分可以督促进度，更加清晰明了整个项目进度。
集中开发测试一起为一个目标努力挺好，我觉得还应该每次提交给测试后，测试整理对应开发人员的测试通过率，就怕有些人真不当团队，

展开

天天向上

2019-11-28

SCM现在还值得认证吗？刚认证了Devops Master

展开